ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.3, March 2008, pp.545−556
DOI: 10.3724/SP.J.1001.2008.00545

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于服务关系本体的服务生成*

徐 萌+, 陈俊亮, 彭 泳, 梅 翔

(北京邮电大学 网络与交换国家重点实验室,北京 100876)

Service Relationship Ontology-Based Web Services Creation

XU Meng+, CHEN Jun-Liang, PENG Yong, MEI Xiang

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

+ Corresponding author: Phn: +86-10-62283130, Fax: +86-10-62285223, E-mail: xumengmoon@gmail.com

Xu M, Chen JL, Peng Y, Mei X. Service relationship ontology-based Web services creation. *Journal of Software*, 2008,19(3):545−556. http://www.jos.org.cn/1000-9825/19/545.htm

Abstract: This paper proposes an interactive and dynamic method—service relationship ontology-based Web service composition method (SROCM). SROCM analyzes the services relationship from three-aspects, presents a method for building the services relationship ontology (SRO). In the composition, SROCM mines the dynamic semantic of services to achieve the interests of users and gets recommend subsequence services by SRO. The semi-automatically method can get an optimized service matchmaking result, adapt for variety of requirements of both customers and service providers flexibly, and ensure efficiency while composing service. The preliminary experiments show that this method can provide excellent subsequent service according to user's interests, and it is feasible and effective in dynamic service composition.
Key words: service association; ontology; service dynamic semantics; service creation

摘 要: 针对服务组合过程中生成复杂业务所涉及的用户个性化、动态生成等问题,提出了一种基于服务关系本体的交互式服务生成方法.该方法从3个视点对服务的关系进行分析,构建了服务关系本体,进而在服务生成中,通过挖掘服务的动态语义,获取用户意图,结合服务关系本体以引导后继服务的匹配.该方法可以灵活地适应用户及业务提供者的需求变化,获得较优的服务匹配结果,同时还提高了服务组合的效率.最后,实现并给出了支持该方法的原型系统以及实验结果,表明了该方法的可行性与有效性.
关键词: 服务关系;本体论;服务动态语义;服务生成
中图法分类号: TP311 **文献标识码**: A

Web 服务作为一种开放的业务提供方式,已经得到了工业界和学术界的广泛认可.现阶段,人们对基于 Web 服务软件系统的集成也越来越关注,目前,手动生成复杂业务的传统方式[1]已不能满足用户个性化及商业逻辑变化的要求,按照用户及业务提供者的需求,动态生成复杂业务是一种灵活、快捷的方法.用户交互式的服务生

* Supported by the National Natural Science Foundation of China under Grant No.60432010 (国家自然科学基金), the National Basic Research Program of China under Grant No.2007CB307100 (国家重点基础研究发展计划(973))

Received 2006-10-14; Accepted 2007-01-24

成方法[2,3]在服务生成中引入用户交互过程,由用户在系统推荐的后继服务中选择所需的服务,执行后,系统再次推荐候选服务,依此类推,直至服务结果满足用户的全部需要.交互式的生成方法避免了自动化服务组合方法[4,5]中对用户需求形式化描述的要求,且可以动态生成复杂业务.此类方法的关键点是服务匹配算法,即如何得到与当前服务相匹配的后继服务,使用户可以在一定数量范围内得到满意的服务,而无须在大量无关服务中进行查找.

为了能使机器理解并自动地处理信息,人们进行了以 RDF[6],OWL[7]为基础的下一代 Web 研究,将 Semantic Web 技术[8]与 Web 服务技术相结合,产生的语义 Web 服务技术使服务语义可以被人机之间、机器之间相互理解.目前,大部分研究都针对利用语义方法对服务的特征进行描述,如 OWL-S[9].我们认为,对复杂业务生成来说,仅对服务个体进行描述是不够的,受多方因素影响的服务关系可对服务匹配结果起决定性作用.同时,服务运行时所具有的特征也还没有被利用,我们发掘了服务运行时的语义以辅助发现用户兴趣点,使得匹配的后继服务体现智能化特点.

本文研究了如何利用服务间的关联关系指导复杂业务的生成,针对信息提供类服务,我们提出了一种用户交互式的服务生成方法——基于服务关系本体的服务生成方法(service relationship ontology-based Web service composition method,简称 SROCM).本方法从 3 个视点对服务关系进行分析,通过领域专家对业务功能的理解、业务提供者对商业策略的规划以及用户使用记录,半自动化地建立服务关系本体.在服务运行过程中,挖掘服务的动态语义,以获取用户关注内容的特点,向用户推荐后继服务,整个复杂服务的生成是服务推荐、选择及执行的统一过程.

本文第 1 节介绍国内外相关工作.第 2 节从 3 个视点对服务之间的关联进行分析和定义,提出服务关系本体及其构建过程.第 3 节给出动态语义、在 SRO(service relationship ontology)中计算语义关联度的算法、SROCM 方法的基本思想及算法分析.第 4 节介绍一个原型系统以及案例分析和实验结果.最后是结论和下一步的研究工作.

1 相关工作

在 Web 服务组合方面,最流行的标准有 BPEL4WS[1],其目标是在过程中将服务的引用从真正的服务实现中抽象出来,这有助于在过程部署或者执行的时候为每个服务选择正确的实现,是一种手动的方法.此外,还有许多针对服务组合自动化的研究,例如,人工智能中的规划和推理技术已被运用到服务组合中,其中,McIlraith 和 Son 提出了基于情景算子的编程语言 ConGolog 来进行服务组合[4];Dan Wu 提出了利用 HTN 方法的 SHOP2 来实现对服务的规划[5].本文所提出的交互式服务组合方法与上述工作的区别在于:SROCM 可体现用户个性化需求,按照客户及业务开发者的需求生成复杂业务,如采用手动生成业务方法,灵活性不够;SROCM 方法无须事先对客户的需求进行形式化描述,由于客户的需求是主观和隐式的,所以,自动化方法中对用户需求进行形式化描述较难实现.Kim 和 Gil 介绍了一个允许用户直接进行交互的服务组合框架[2],用户可利用已有的模板进行一定程度的服务选择.对于用户参与的思想,我们与其基本类似,区别在于,我们把服务的执行、推荐、选择作为一个统一的服务组合过程,即用户可以在执行完上步服务后,根据上步服务的结果,在推荐的服务中进行选择.

服务组合的关键问题是如何针对当前服务找到匹配的后继服务.目前,基于接口的语义匹配方法研究较多[3,10,11],此类方法从服务间数据交换出发,以服务接口相匹配作为匹配标准.但接口匹配方法搜寻服务的过程复杂而庞大,且仅靠接口的匹配往往会得到一些用户不需要的结果,导致用户选择困难.文献[3]在服务描述中添加参数,利用服务参数以及用户的 QoS(quality of service)要求进行过滤,可缩小后继服务范围,但用户需对每项服务参数进行设置.文献[12]提出了按领域组织服务的思想,在服务注册时对服务领域进行归类,利用 P2P(point to point)进行服务查找,可以提高服务的匹配效率,但是没有利用服务的领域特色并用以指导服务匹配.以上文献中的接口匹配方法都是基于服务的接口描述进行,而无法体现服务运行时的特点.本文提出的服务动态语义,是指服务运行时所具有的语义,利用服务动态语义可以获取用户关注内容的特点,发现服务潜在的语义,用于服务匹配算法可以提高匹配结果质量.

服务接口匹配方法具有自动化特点,从服务描述可以得到服务之间的数据关联,但无法将服务组合的其他影响因素考虑在内,如用户习惯和商业逻辑等.一些研究者初步讨论了服务之间的关联对服务生成的影响[13,14],文献[13]发掘了根据服务的前置条件及后置结果得到服务执行的顺序关系,包括并行、顺序、替代等关系,但是仅考虑了服务执行顺序关系,而未对服务之间的功能关系进行分析,且不适用于没有前置条件和结果的信息提供类 Web 服务;文献[14]提出了建立领域任务本体以动态生成工作流的方法,这里的领域本体结构类似于 HTN 任务层次网络[9],因此也保留了规划方法的缺点,对于业务逻辑修改的适应能力较差,也无法体现用户的个性化特点.与这些方法相比,本文分析服务关系时综合考虑了服务功能具有的领域特性、商业策略以及用户行为习惯的特点,从 3 个视点获取服务间的关联关系,进而利用其辅助服务组合,可以体现智能化特点,充分满足了用户及业务提供者多方面的要求.

2 服务间关联关系

我们认为,对服务个体的描述不能刻画服务之间的关联关系,而对于服务组合来说,服务间的关系对匹配结果起着决定的作用.为了实现 Web 服务的发现和组合,本文从 3 个视点定义了服务之间可能具有的关系,进而生成服务关系本体.

定义 1. 领域本体 O.Web 服务组合所需的概念及概念之间关系的集合,可用四元组表示 $O=\{C,R,rel,A\}$.其中,C 表示概念集合,R 表示关系集合,rel 表示概念间的关系,A 表示本体公理.如图 1 所示的地理场所本体片断,Web 服务的 OWL-S[9]中的概念来自集合 C.

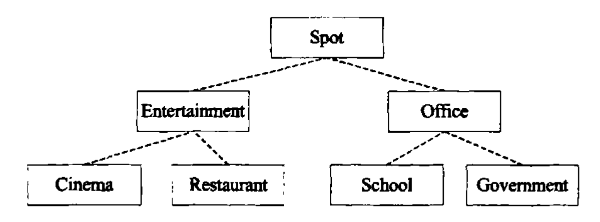


Fig.1 The snippet of geography ontology
图 1 地理场所本体片断

2.1 服务关系的多视点分析

在现有的商业运营模式中,我们认为复杂服务的生成过程包括以下 3 个主要角色:服务、第三方服务提供者(下文简称 3rdSP)和用户.其中,服务负责完成一定的任务功能,3rdSP 负责集成多个服务以生成具有商业逻辑的复杂业务,而用户可能参与服务的组合过程并最后享受复杂服务.服务间存在的关系受这 3 个角色的影响,因此,我们提出从 3 个视点对服务关系进行分析,这 3 个视点分别为服务的主题视点、商业逻辑视点和用户行为视点.这里用描述逻辑(description logic)表达服务关系,以地理场景相关服务为例进行讨论.

2.1.1 服务主题视点

定义 2. 服务主题是服务功能所面向的领域,用集合 T 表示,$T\subset C$.主题具有领域特点,即相关系统的共性和特性.如餐饮领域中,查询人均消费服务是所有餐饮场所共性的需求,而不同的餐饮场所,如餐厅与酒吧,分别具有查询特色菜服务和查询演出内容服务,则体现出了各子领域的特性.服务之间借助服务与主题之间以及主题之间的关系形成如下关联关系:

- 继承关系,表示主题之间的"is-a"关系,用 $subClassOf(T_i,T_j)$表示.例如,主题电影院与主题娱乐场所之间的关系为继承关系 R_1:$Cinema\subseteq\forall subClassOf.Entertainment$.
- 从属关系,表示服务与主题之间的"part-of"关系,一个服务可以有一个或多个主题,用 $hasTopic(T_i,S_i)$表示.例如,查询影讯服务和影院具有从属关系 R_2:$queryFileInfo\subseteq\forall hasTopic.Cinema$,查询人均消费服务和娱乐场所具有从属关系 R_3:$queryAverage\subseteq\forall hasTopic.Entertainment$;由 R_1 和 R_3 可推出 $queryAverage\subseteq$

$\forall$*hasTopic.Cinema* 为真,即查询人均消费与电影院有从属关系.

主题关联思想类似于信息资源导航中的主题映射(topic maps)概念[15],根据服务功能的内容对其按领域进行分类.其不同点是,在主题映射中,任意概念都可以作为主题,以对复杂的信息建立索引.而这里的主题是服务的核心内容,并且主题间具有继承关系,子领域具有父领域的服务,同时具有个性服务.服务主题视点表现服务功能之间的联系,是服务关系的基础视点.

2.1.2 商业逻辑视点

各个 3rdSP 随着市场变化,需要调整其商业策略以吸引更多用户,因此,商业逻辑视点表达 3rdSP 对服务商业逻辑、策略的需求.目前,依据信息提供类服务的需要,商业逻辑可产生以下 3 种服务关系:

- 一步执行关系,用 *pushTo*(S_m,S_n)表示,表达商业策略如下:3rdSP 为了给用户提供方便,用户调用服务 S_m 之后,则系统自动触发执行 S_n.例如,执行查询地址服务后,自动触发显示周围地图服务,则存在关系 *queryAddress*$\subseteq\forall$*pushTo. ShowAroundMap*.
- 依赖关系,用 *dependOn*(S_m,S_n)表示,表达只有服务 S_m 执行的情况下,才可以执行服务 S_n.例如,执行发送短信服务必须以用户登陆服务为前提,则存在关系 *sendAMessage*$\subseteq\forall$*dependOn.login*.
- 互斥关系,表示为 *complementOf*(S_n,S_m),表达如果服务 S_m 被调用后,S_n 不会被调用.例如,假设设置查询自驾游路线服务 *queryAutoRote* 与查询公交换乘路线服务 *queryBusTransfer* 为互斥关系,当调用 *queryAutoRote* 以后,则不再推荐 *queryBusTransfer* 服务.

2.1.3 用户行为视点

用户行为视点是从用户使用服务的习惯出发,当系统积累大量用户使用记录后,可以通过机器学习技术发现服务调用之间的内在关系,我们总结为两种递进关系:

- 主题递进关系,表示某个主题与服务之间有较强的推荐关系,用 *adviceTo*(T,S)表示.例如,当用户调用查询地址服务时,对于输入某编辑部的用户,选择的后继服务可能为查询邮编 *queryMailCode* 或邮箱地址服务 *queryMailAddr*;而对于输入某景点的用户,选择的后继服务往往为查询天气服务 *queryWeather*,而不关心邮编或邮箱地址.这里的主题与服务之间具有主题递进关系,即

  queryWeather$\subseteq\forall$*adviceTo.SightSpot*;*queryMailCode*$\subseteq\forall$*adviceTo.Newsroom*.
- 服务递进关系,表示两个服务之间有较强的推荐关系,这里的两个服务多为同主题的服务,表示为 *suggestTo*(S_m,S_n).例如,用户习惯在查询地址 *queryAddress* 后查询换乘信息 *queryTransferInfo*,这里,两个服务之间存在递进关系,即 *queryTransferInfo*$\subseteq\forall$*suggestTo.queryAddress*.

2.2 服务关系本体

定义 3. 领域服务关系本体(service relationship ontology,简称 SRO)定义为一个四元组$\{T,S,R,R_C\}$,其中,T 表示服务的主题集合;S 表示可用服务集合,每个可用服务 S_i={*Name*,*Occurrence*},其中,*Name* 表示服务名称,*Occurrence* 表示服务与具体服务描述文件的位置;R 表示服务及主题之间的关系,$R\in\{T\times T,S\times S,T\times S\}$.由第 2.1 节定义可得,$R$={*subClassOf*,*hasTopic*,*pushTo*,*adviceTo*,*suggestTo*,*dependOn*,*complementOf*}.

我们分 3 个阶段半自动化地构建 SRO:

(1) 首先是服务主题视点,主题的概念关系来自于领域本体,构成 SRO 的基本骨架,通过对现有服务的接口描述进行分析,手动将服务定位于多个主题中;(2) 其次是商业逻辑视点,我们定义了基于 SWRL 的商业策略描述模型,由 3rdSP 定制、添加服务之间的策略,系统可自动转换为服务关系,详细介绍见文献[16];(3) 最后是用户行为视点,在系统运行过程中记录用户行为,通过机器学习的方法挖掘服务之间及主题与服务之间潜在的关联,如果服务之间或者主题与服务之间存在统计上的连续选择规则,则半自动地添加此类服务关系.其中,服务主题体现了服务自身功能所产生的关联关系,具有一定的通用性和可重用性.而用户行为视点和商业逻辑视点对于不同的 3rdSP 及应用场景,个性特点突出.

SRO 在独立的服务之间建立了服务关联,可能会引入一些语义重复和矛盾.因此,针对生成的 SRO,必须进行语义的一致性检查及精简工作,否则,在复杂业务的生成过程中会引起矛盾或失败.一致性检测主要对于 SRO

中的关系进行循环检测和冲突检测:循环检测主要检测引起系统自动执行的关系,如 *pushTo* 关系,如果存在 $S_a \subseteq \forall pushTo.S_b$ 和 $S_b \subseteq \forall pushTo.S_c$,则不能存在 $S_c \subseteq \forall pushTo.S_a$,以避免出现系统执行进入死循环状态;冲突检测对具有语义约束的关系对进行检查,检查是否两个服务间同时存在以下关系:(*dependOn,complementOf*),(*suggestTo,complementOf*) 或 (*pushTo,complementOf*).例如,如果存在 $S_a \subseteq \forall dependOn.S_b$,则不能存在 $S_a \subseteq \forall complementOf.S_b$.总之,通过自动化的属性一致性检测,可实现对 SRO 的一致性检查.

3 基于服务关系本体的服务生成方法

在交互式的服务组合方法中,系统向用户提供可选的后继服务,由用户进行选择.因此,核心算法是后继服务的匹配算法,即如何向用户提供符合用户兴趣的后继服务.本节介绍了服务动态语义、在 SRO 中计算服务之间语义关联度算法以及后继服务匹配算法和算法分析.

3.1 服务的动态语义与静态语义

定义 4(服务). 一个服务可用下面的表达式来描述 $S_i(I_i,O_i)$,其中,$I_i \subset C$,表示服务 S_i 需要的输入参数集合(input);$O_i \subset C$,表示服务 S_i 在调用成功后输出的参数集合(output).

定义 5(服务静态语义(static semantics,简称 SS)). 利用领域本体中概念对服务属性进行的描述,表示为 $S_i.SS=\{I_i.SS,O_i.SS\}$,依据目前对服务描述的定义,$S_i.SS=S_i$.

定义 6(服务动态语义(dynamic semantics,简称 DS)). 服务执行时,输入参数和输出参数所具有的语义,它在服务执行时产生,因此称其为动态语义,表示为 $S_i.DS=\{I_i.DS,O_i.DS\}$,其中,$I_i.DS \subset C,O_i.DS \subset C$.

动态语义主要特征为会话特征,即服务的动态语义只在与一个用户的当前会话中有效.进一步分析,在一个会话内,*DS* 受以下因素影响:(1) 时限因素.*DS* 在服务被调用时产生,保存到服务下一次被调用之前.比如,用户先调用查询地址服务,查询内容为“小西天电影院”,此时服务的 *DS* 为 Cinema,然后又再次调用此服务,查询内容为“北京邮电大学”,此时服务的 *DS* 为 School;(2) 空间因素.当用户在不同地理位置调用服务时,服务动态语义具有地理位置信息特点.比如,用户在家中调用与在风景区调用服务,会话保存的服务动态语义不同;(3) 资源因素,用户触发服务的工具特点.比如是固话接入、手机接入或者其他接入方式.通常情况下,空间因素、资源因素是与会话绑定的,即一个会话中,这两个因素引起的动态语义不会变化;而在一个会话中,时限因素引起的动态语义可能为多个.本文讨论以时限因素所产生的 *DS* 为主,由于服务通常面向同一类对象,具有通用特性,因此,服务接口描述也具有一定的抽象特性,从而大部分服务具有丰富的 *DS*.

动态语义与静态语义的关系有 3 种(详细举例见表 1):

① $S_i.DS \subseteq S_i.SS$,即 *DS* 是 *SS* 的具体和深化(如 S_1);

② $S_i.DS=S_i.SS \cup N$,N 表示新的概念集合,即 *DS* 对 *SS* 赋予新的语义(如 S_2);

③ $S_i.DS=S_i.SS$,即在服务接口描述的一定程度上,*DS* 还可与 *SS* 相同(如 S_3).

本文讨论以时限因素所产生的 *DS* 为主,由于服务通常面向同一类对象,具有通用特性,因此,服务接口描述也具有一定的抽象特性,从而大部分服务具有丰富的 *DS*.

Table 1 Relation list of DS and SS

表 1 服务动态语义与静态语义关系列表

Type of relation	Service *S*	Factcr *F*		*DS*	*SS*
$DS \subseteq SS$	QueryAddress service S_1	*F*=time	Input value="Beijing University of Posts and Communications"	School	Spot
$DS=SS \cup N$	CallForwarding service S_2	*F*=time	62283320 (The phone No. of Xiaoxitian Cinema)	Cinema	TelephoneNo
		F=location	User's in home	Home	TelephoneNo
		F=resource	Call by mobilephone	Mobile	TelephoneNo
$DS=SS$	QueryFeature service S_3	*F*=time	The restaurant of Tianwaitian	Restaurant	Restaurant

3.2 服务之间的语义关联度

服务间语义关联度体现服务之间语义关联的紧密程度,可用本体中实例间的语义距离来表达.语义距离是在同一个本体中,对两个不同的实例之间存在的继承关系或者二元关系链的一种度量.有关语义距离应用较多的算法在文献[17]中已有总结.在SRO中,我们以服务主题之间的语义距离来评价服务的关联度,因此,我们通过加入服务间多主题的继承关系对GSCM距离算法进行了扩展.再者,当前服务的主题由其动态语义决定,例如,查询地址服务,*I.SS=Spot*.当服务运行时,如果输入参数为"小西天影院",则我们可以得到服务的动态语义 *I.DS=Cinema*,此时,$queryAverage \sqsubseteq \forall hasTopic.Cinema$.服务的动态语义来自服务运行时具体的输入参数值,它的获取需利用现有的分词技术及本体概念映射技术,这里采用中国科学院ICTCLAS分词系统[18]对输入信息进行分词,然后将分词结果映射到C的相应概念.服务动态语义关联度算法见算法1.

算法1. 服务动态语义关联度算法.

输入:服务的URI,分别为uri_1,uri_2;

输出:*assDegree*,服务之间语义关联度.

1) $T_1=getTopic(uri_1)$;　　　　//根据uri_1动态语义获取主题集合

　$T_2=getTopic(uri_2)$;　　　//获取uri_2的主题集合

　$T_1=\{T_{11},T_{12},\ldots,T_{1m}\},T_2=\{T_{21},T_{22},T_{23},\ldots,T_{2n}\}$

2) 初始化$D[m][n]$中每个值为0;

```
    For (i=0;i<m;i++) {
    For (j=0;j<n;j++) {
        If (T1i==T2j) {D[i][j]=1;}
    else {D[i][j]=1/getGCSM(T1i,T2j)}}
    }
```
　//计算每个主题间GCSM距离,如果*topic*相同,$D=1$;否则,$D=1/GCSM(T_1,T_2)$.

　$D_{com}=\sum_{i=1,j=1}^{i=m,j=n} Dij$,　　//$D_{com}$为所有*topic*语义相关度之和

3) $D_{diff}(T_1,T_2)=T_1\cup T_2=m+n$　　//D_{diff}为主题间差异

4) $assDegree=D_{com}/D_{diff}$　　//相关度为相同系数与差异系数之商

5) 返回*assDegree*

6) End.

说明:*getGCSM*()为GCSM算法,算法代码见文献[15].

算法中每两个服务之间需要计算GCSM距离的次数为两个服务主题数量的乘积,所以,算法复杂度为$O(m\times n)$,由于m,n值为较小的正数,因此算法复杂度并不高.由于语义距离的频繁计算会导致整个算法性能下降,可采用"缓存"技术[17]预先对SRO中每对服务之间的语义距离进行离线计算并保存.

3.3 SROCM服务匹配算法

在SRO中,不同关系在服务组合中的作用不同:服务主题视点体现服务领域特征,可用于确定服务搜索空间和判断服务间功能关联的紧密程度;商业逻辑视点对服务设定限制,使服务按照一定规则执行,可用于对匹配服务的过滤及补充;用户行为视点所产生的服务关系直接指明服务之间的连接关系,此关系最有可能被用户选择到.服务匹配算法的核心思想是:① 根据当前服务curS的动态语义在SRO中定位服务,依据服务主题获取服务搜索空间,在搜索空间内,利用接口匹配的方法进行服务匹配;② 找到与curS有直接关系的对象,再根据商业逻辑视点的关系对匹配结果进行过滤和扩展;③ 将匹配结果中的服务按照其与curS的语义相关度降序排序,得到最后的服务匹配结果.文献[4]中详细定义了接口之间的语义匹配关系,只有在nextS与curS接口参数类型的关系属于概念相等或泛化关系这两种情况下,服务才是可匹配的.SROCM的服务匹配见算法2.

算法 2. SROCM 的服务匹配算法.

输入参数:*preS*:前步执行服务,*context*:用户 *context*;

输出参数:后继服务列表,待执行列表.

1.1 根据 *preS* 动态语义,得到其当前主题 *topic*,在 SRO 中设置其空间扩散步数 *steps*=*n*,则找到该 *topic* 的第 *n* 个祖先作为搜索空间的 *Topic*,得到属于此 *Topic* 的所有服务,生成搜索空间$\Omega=\{s_1,s_2,\ldots,s_p\}$;

1.2 在Ω内,遍历每个服务,利用上一步输出 *O*(*preS*)以及用户 *context* 中结果组合作为目标,找到输入参数与以上参数及参数组合相匹配的服务,放入匹配列表 $Next_{IO}.associationDegree(S_c,Next_{IO}[i])$,得到语义关联度 *assDegreeList*;

2.1 根据 *Topic*,得到 *Topic* 所有 *contains* 属性的对象值,放入 $Next_{SRL}$,设置语义关联度为α=0.9;

2.2 如果 *preS* 具有 *adviceTo* 属性,把其对象放入 $Next_{SRO}$,设置语义关联度为β=1;

2.3 如果 *preS* 具有 *pushTo* 和 *dependentOn* 属性,把属性的对象加入待执行列表 *toDoList*;

3.1 $Drop(Next_{IO},Next_{SRO})$,

3.2 *NextService*=*OrderByMaopao*(*assDegreeList*);

说明:

$Drop(list_1,list_2)$:过滤 $list_1$ 中与 $list_2$ 相同的服务,保留其较大的关联度;

去掉 $list_1$ 中与 *preS* 有 *compelentOf* 的服务;

OrderByMaopao(*list*)应用冒泡算法进行降序排序;α及β的值可以根据关系的重要性来设置.

3.4 算法特点

(1) 利用服务动态语义确定用户关注领域,在一定范围内进行 I/O 接口匹配,可过滤掉与用户关注内容不相关的领域,合理缩小了服务的搜索空间.例如,当 cruS 的动态语义为影院 Cinema 时,可将主题扩散到娱乐场所 Entertainment,而不考虑办公场所 Office 域空间,因为当用户关注电影院时,再选择办公相关服务的可能性极小.

(2) 利用服务的动态语义可体现用户关注内容的特点;加入了服务关系的推荐,根据 SRO 对 I/O 匹配结果进行添加和过滤,既体现了推荐服务智能化的特点,又使得复杂业务的逻辑结构可由商业策略来控制,此特点使 SROCM 的商用化成为可能.

(3) 用户选择时,面临的服务是根据语义关联度排序后的结果,与用户关注内容紧密的服务排列在前,使用户可以较快地发现所需后继服务.

4 方法实现及评价

我们设计的"综合智能移动业务平台"[19]实现了基于 SROCM 方法的服务生成系统,业务平台提供的基本服务包括 GIS 服务、电信服务和其他内容的信息服务,可生成综合信息类的电信增值业务.平台引入话务员辅助用户进行服务的生成和执行,用户只需接通电话即可得到满意的个性化服务.复杂服务生成及应用场景如下:用户拨打特服号码接入话务员,由话务员负责与系统交互,向用户提示下一步可以执行的服务,用户选择服务后,话务员通过平台执行业务,如此引导和辅助用户完成复杂业务.

4.1 业务生成系统框架

我们的实验系统体系结构如图 2 所示,其中包括知识库、用户 Portal、业务生成模块(service creator)以及业务执行模块(service executor) 4 部分.

知识库中包括 3 个本体库:

(1) 服务关系库(service relation ontology library),存放领域内服务关系本体,使用 OWL-DL 描述;

(2) 领域本体库(domain ontology library),存放基本概念集合 *C*,用于基本的语义理解;

(3) 服务库(service library),存放可用业务的语义描述,我们使用 OWL-S 语言描述服务.

这 3 个本体库之间具有依存关系,其中,SRO 描述服务关系,同时保存了服务实例到服务库之间的映射;SRO

及服务库中的概念及概念关系来自于领域本体库.

用户 Portal 负责维护用户的会话,获取用户输入参数的动态语义信息.User Context 保存一次会话中,用户与系统交互的所有数据.

Service Creator 是业务生成的核心模块,实现服务匹配算法.

Service Executor 具有服务调用、返回生成结果、维护 User Context 等功能.

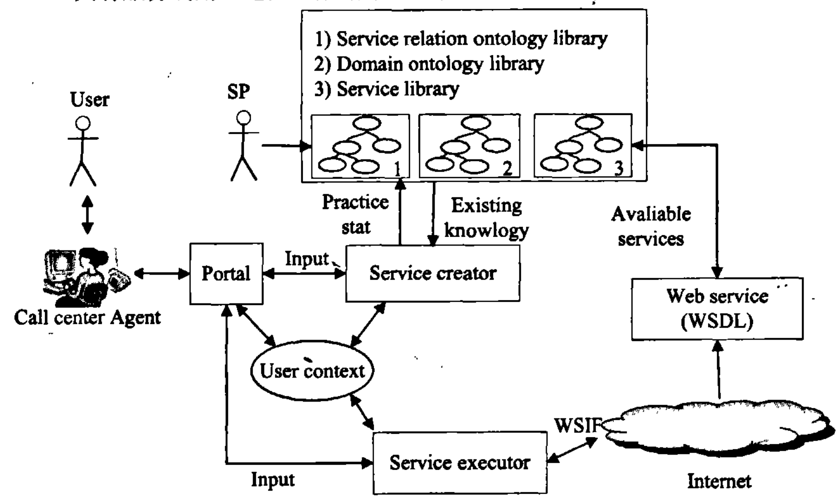


Fig.2 The framework of SROCM

图 2 基于服务语义关联的业务生成框架

4.2 案例分析

本文以地理位置相关信息查询服务为主构建 SRO,与案例相关 SRO 的片断如图 3 所示.

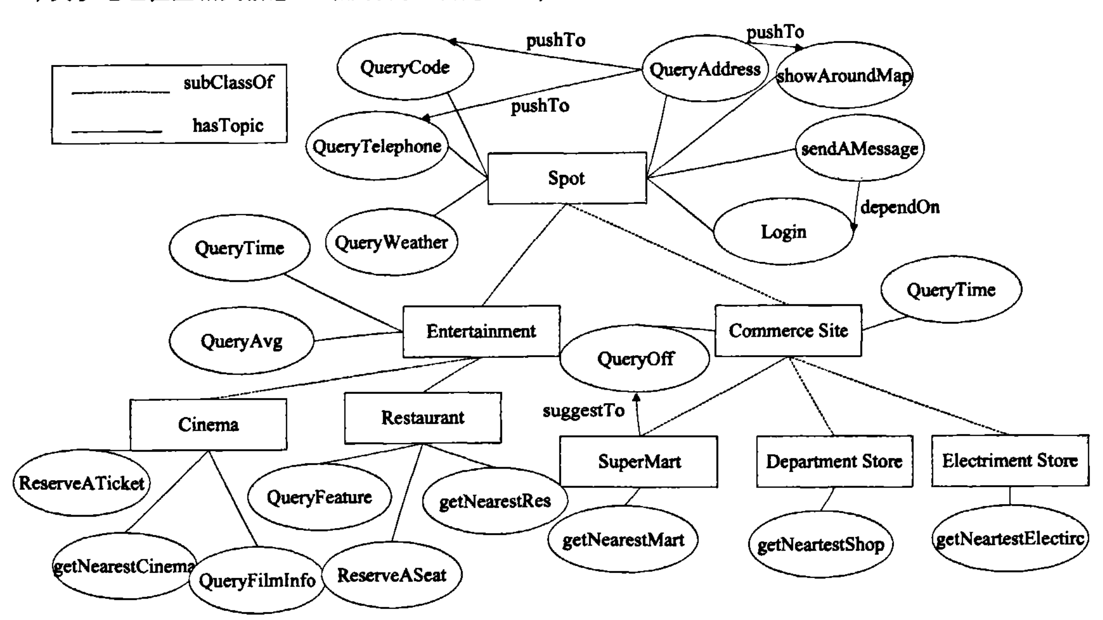


Fig.3 The snippet of SRO

图 3 SRO 本体片断

平台的话务员工作界面如图 4 所示,右边深色框内上部分为领域选择,下部分为向用户推荐的后继服务.一个典型案例如下:当用户电话接入后,话务员提示基本服务,如查询地点服务、交通换乘服务等.用户选择查询地点服务,话务员执行该服务.当执行查询地址服务后,平台同时执行查询电话、显示周边地图服务,结果如图 4,并在右边深色框内显示推荐的候选服务.用户主动提出或者话务员向用户提示下一步可选服务.按用户要求,话务员依次完成以下任务:查询当前位置到影院的换乘路线,把换乘路线地图发送到用户手机.



Fig.4 The GUI of SROCM for call center Agnet
图 4 话务员工作界面

下面以两个应用场景为例,分别验证 SROCM 匹配服务算法推荐服务情况及业务生成系统适应商业逻辑变化的能力.

服务场景 1 用于验证服务匹配效果,对案例中某步服务的匹配结果进行分析.当用户查询地点服务时,输入"五道口电影院"之后,后继服务列表与传统接口语义匹配方法的结果比较如下:

目前服务库中,针对查询地址服务,基于 I/O 接口匹配的方法返回所有输入为 Address 的服务:查询附近饭店、查询附近电影院、查询附近酒吧、查询附近商场、查询附近加油站、查询附近停车场、查询附近银行、查询附近律师事务所、查询附近政府部门、查询附近学校、查询附近医院、查询附近邮局、发送信息、查询换乘信息等共 21 项服务.

SROCM 方法得到的结果为:当 *Steps*=1 时,查询影讯信息、查询影院情况、查询预定电影票、查询人均消费、查询附近影院、查询附近剧院、发送信息、查询换乘信息、查询附近加油站、查询附近停车场、查询附近银行共 11 项服务;当 *Steps*=2 时,比 *Steps*=1 时的结果增加了附近饭店、附近酒吧两项服务.SROCM 方法还可以自动触发服务:查询周围地图服务.

可以看出,I/O 匹配方法的结果中查询附近律师事务所、查询附近政府部门、查询附近医院、查询附近邮局等服务,都与用户目前关注的影院相差较远;而 SROCM 方法除了将服务范围限定在一定娱乐相关范围内外,还可以辅助得到查询影讯、预订电影票等相关主题的服务.如此,用户可以在较少的选择范围内得到与前继服务相关度较高的服务.

服务场景 2 用于验证系统动态适应商业逻辑更改的能力,例如,3rdSP 需设定以下商业规则:

(1) 假设当用户关注超市这个主题之后,平台推荐此超市的打折信息服务.

(2) 用户发送短信之前,需要进行身份验证.

通过 SWRL 语言对商业规则的描述,系统可转换为 SRO 中的以下断言:

(1) $queryOffInfo \subseteq \forall suggestTo.SuperMarket$

(2) $login \subseteq \forall dependOn.sendAMessage$

复杂业务的生成过程是:用户选择了查询地址服务,输入参数为“天客隆超市”,则推荐服务除了包括商场Shop范围内接口匹配的服务之外,还包括查询商场打折信息服务;当用户选择查询打折信息后,要求把信息以短信形式发送到用户手机时,系统自动检查用户是否已经登陆.

对以上两个应用场景进行分析后可以看出:SROCM 匹配算法可以减少匹配服务的数量,同时可以提高匹配结果的满意度,明显优于传统的基于接口的语义匹配方法;且对于业务逻辑的修改简单易行,可灵活地适应商业逻辑的变化;该方法从服务的一般特性出发,考虑服务、用户及 3rdSP 角色观点,与业务本身无关,具有良好的扩展性.

4.3 性能评价

在原型系统基础上,我们对算法的性能进行验证,实验环境是:CPU 为 Intel Pentium IV 2.8GHz,内存为512MB,操作系统为 Windows XP,算法实现语言为 Java.鉴于目前没有相关的标准平台和标准测试数据集,本文对现有的服务关系本体(如图3所示)按结构比例对服务进行复制,生成模拟Web服务作为测试用例.我们共生成5 个测试数据集,所包含的服务总数分别为 200,400,800,1 200 和 1 600.

实验 1 用于验证与接口匹配方法(IO Matchmaking)相比,SROCM 对服务搜索空间的限制.我们在层次 2 和层次 3 中各抽取 20 项服务作为 *preS* 进行验证,均设置 SROCM 方法的主题扩散步数 N 分别为 1,2,3 时,得到的服务平均搜索空间与原始空间的比例如图 5 所示.由于目前 SRO 最深层次为 3,可以看出,当服务层次为 2 时,扩散一步可以缩小空间 77%,扩散两步则达到原始空间;而当服务层次为 3 时,可分别扩散 1~2 步,使服务搜索空间分别减少 59%和 77%.扩散步数大,空间随之变大.我们对搜索空间进行限制,过滤的空间为与用户关注领域无关的部分,因此,SROCM 算法对空间的限制是有效的,既过滤掉了不相关的服务,又可以提高服务搜索的效率.

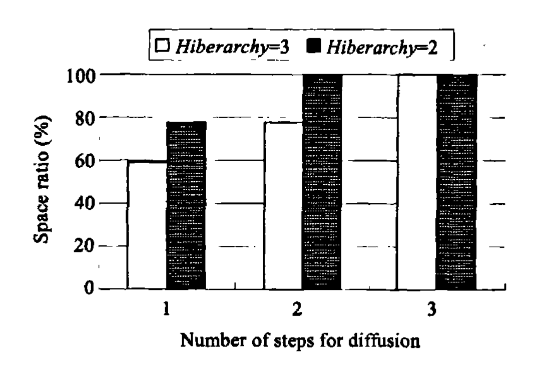


Fig.5 Comparison of service searching spaces between SROCM and I/O matchmaking

图 5 服务搜索空间对比

实验 2 用于验证 SROCM 方法匹配服务的效率.我们从 6 个测试集中各抽取处于层次 3 的 10 项服务作为 *preS*,分别用 SROCM 算法(*steps* 为 1 步、2 步、3 步)和 I/O 接口匹配方法求得后继服务,以获得匹配消耗的时间取均值作为结果,对 5 个测试集所用时间(单位:ms)及时间比例见表 2.可以看出,SROCM 方法中,当 *Step*=1 和 *Step*=2 时,消耗时间小于 I/O 接口匹配方法;但当 *Step*=3 时,消耗时间大于 I/O 接口方法.我们对 5 组数据分别求其时间比例,取均值,列出时间和空间对比,见表 2.理论上,算法匹配时间的缩小应该与空间减少成正比,但从表 2 可以看出,扩散 1 步、2 步所消耗时间的减少没有达到空间减少的比例,扩散 3 步后比 I/O 匹配消耗时间增涨至128.6%,可见,压缩空间并不能带来相应比例的性能提高.由于采用了“缓存”技术,所以语义距离部分对性能的影响不大,其算法时延主要由对 SRO 的逻辑推理所致.但是,在空间扩散一步到两步的情况下,由于搜索空间得到了合理的限制,服务匹配总时间有所提高.且当服务总数为 1 600 时,*Step*=2 时的每次匹配消耗的时间为 365ms,是用户可以容忍的时间.

从以上实验可以看出,算法扩散步数对服务匹配的效率影响较大,如果空间关联度设置过小,则有可能得不

到用户需要的后选服务;如果设置过大,则失去用户关注点的意义,并引起性能下降.根据现有 SRO,我们验证得知,当服务位于较高层次时,扩散步数可以为 1 或 0;而位于较低层次时,扩散步数可以为 2,此时既能满足用户对服务的需要,又能保证一定的匹配效率.

通过案例分析及性能评价可知,SROCM 算法可得到与用户关注点相关同时满足提供者商业逻辑的后继服务,且能有效地限制服务搜索空间,在保证服务匹配质量的情况下,高效地进行服务组合,表明了服务组合的可行性及有效性.

Table 2 Comparison of delay between SROCM and I/O matchmaking

表 2 消耗时间比(I/O 匹配/SROCM)

Delay(ms) / Method / Dataset	I/O matchmaking	SROCM (*Step*=1)	SROCM (*Step*=2)	SROCM (*Step*=3)
200	50	29.1	45.1	62
400	161.2	97.8	155.2	211.2
800	181.6	107.2	174	254.3
1200	281	172.1	264	391.2
1600	408.1	242.8	365	569.6
Proportion of average delay (%)	100	61.5	94.0	128.6
Proportion of average searching space (%)	100	59	77	100

5 总 结

现有服务组合方法多基于对服务个体的描述进行服务匹配,且多停留在服务静态语义层面上,难以发现用户关注的特点,限制了服务动态组合的能力.本文首先讨论了服务间的关联关系,从 3 个视点对服务关联进行分析,从多方面视点构建 SRO,提出了基于服务关系本体的交互式服务生成方法,该方法将服务关系应用于服务匹配过程;进而,我们提出了服务运行时的动态语义,可以准确地发现用户的兴趣特点,以进行服务的推荐;最后,我们把服务的执行、推荐、选择过程结合为一个统一的过程,使得用户在参与服务组合的时候,可以通过上一步服务的结果选择后继服务.SROCM 可在缩小用户选择范围的同时,给用户提供较优的候选服务.对于业务开发者来说,只需维护服务关系本体库,就可进行商业逻辑控制、修改生成业务的流程等操作,因此,本方法可以灵活地适应业务提供者商业策略的变化.本文实现的原型系统及实验,验证了 SROCM 算法的可行性与有效性.下一步工作中,针对 SRO 中对关系间的约束,可使用扩展的描述逻辑 $ALC(\neg)$[20]进行推理以进行一致性检测,但是目前,对于此扩展描述逻辑目前尚无推理机支持,这也是我们今后研究的方向.

References:

[1] BPEL4WS Consortium. Business process execution language for Web services. 2002. http://www.ibm.com/Developerworks/library/ws-bpel

[2] Kim J, Gil Y. Towards interactive composition of semantic Web services. Technical Report papers. Palo Alto: AAAI, 2004. 100–107. http://www.isi.edu/~jihie/papers/SSS04Kim.pdf

[3] Sirin E, Parsia B, Hendler J. Filtering and selecting semantic Web services with interactive composition techniques. Intelligent Systems, 2004,19(4):42–49.

[4] McIlraith S, Son T. Adapting golog for composition of semantic Web services. In: Fensel D, Giunchiglia F, McGuinness D, Williams MA, eds. Proc. of the 8th Int'l Conf. on the Principles of Knowledge Representation and Reasoning. Toulouse: Morgan Kaufmann Publishers, 2002. 482–496.

[5] Wu D, Parsia B, Sirin1 E, Hendler1 J, Nau D. Automating DAML-S Web services composition using SHOP2. In: Fensel D, Sycara K, Mylopoulos J, eds. Proc. of the 2nd Int'l Semantic Web Conf. LNCS 2870, Sanibel Island, 2003. 92–107.

[6] Manola F, Miller E. RDF primer. 2004. http://www.w3.org/TR/REC-rdf-syntax/

[7] Antoniou G, van Harmelen F. Web ontology language: OWL. In: Staab S, Studer R, eds. Handbook on Ontologies in Information Systems. Springer-Verlag, 2003. 67–92.

[8] McIlraith SA, Son TC, Zeng HL. Semantic Web services. IEEE Intelligent Systems (Special Issue on the Semantic Web), 2001, 16(2):46–53.

[9] Martin D, Burstein M, *et al.* OWL-S: Semantic markup for Web services. 2003. http://www.daml.org/services/owl-s/1.1/overview/

[10] Li M, Wang DZ, Du XY, Wang S. Dynamic composition of Web services based on domain ontology. Chinese Journal of Computers, 2005,28(4):644–650 (in Chinese with English abstract).

[11] Oh SC, On BW, Larson EJ, Lee D. BF*: Web services discovery and composition as graph search problem. In: Werner B, ed. Proc. of the e-Technology, e-Commerce and e-Service. Hong Kong: IEEE, 2005. 784–786.

[12] Arpinar IB, Zhang RY, Aleman-Meza B, Maduko A. Ontology-Driven Web service composition platform. In: Bichler M, Chung JY, eds. Proc. of the Int'l Conf. on E-Commerce Technology. San Diego: IEEE, 2004. 146–152.

[13] Lin L, Arpinar IB. Discovering semantic relations between Web services using their pre and post-conditions. In: Cantarella JD, ed. Proc. of the Int'l Conf. on Service Computing. Orlando: IEEE, 2005,2:237–238.

[14] Chun S, Atluri V, Adam N. Domain knowledge-based automatic workflow generation. In: Hameurlain A, Cicchetti R, eds. Proc. of the 13th Int'l Conf. on Database and Expert Systems Applications. LNCS 2453, Aix-en-Provence, 2002. 81–92.

[15] Michel B, Martin B, Newcomb Steven R. 1999, ISO/IEC 13250 topic maps: Information technology—Document description and markup languages. http://www.topicmaps.org/xtm/

[16] Xu M, Chen JL. Integrating semantic business policy into Web service composition. In: Ahrens M, Schonherr M, eds. Proc. of the 1st Int'l Workshop on Service Oriented Modeling. Berlin: GITO-Verlag, 2006. 84–95.

[17] Zhang P. The research and implementation of semantic based web services discovery [MS. Thesis]. Beijing: Tingshua University, 2005 (in Chinese with English abstract).

[18] Zhang HP, Yu HK, Xiong DY, Liu Q. HHMM-Based Chinese lexical analyzer ICTCLAS. In: Walker DE, ed. Proc. of the 2nd SigHan Workshop. Sapporo: ACM, 2003. 184–187.

[19] Xu M, Meng XW, Chen JL, Mei X. The research of load balancing for integrated service platform. Journal of Beijing University of Posts and Telecommunications, 2006,29(5):94–97 (in Chinese with English abstract).

[20] Lutz C, Sattler U. The complexity of reasoning with Boolean modal logic. In: Advances in Modal Logic, Vol.3. Stanford: CSLI Publications, 2001.

附中文参考文献:

[10] 李曼,王大治,杜小勇,王珊.基于领域本体的Web服务动态组合.计算机学报,2005,28(4):644–650.

[17] 张钋.基于语义的网络服务匹配机制的研究与实现[硕士学位论文].北京:清华大学,2005.

[19] 徐萌,孟祥武,陈俊亮,梅翔.综合业务平台负载均衡算法的研究.北京邮电大学学报,2006,29(5):94–97.



徐萌(1979－),女,山东青岛人,博士生,主要研究领域为Semantic Web Service,动态组合技术.



彭泳(1978－),男,博士,副教授,主要研究领域为Semantic Web技术.



陈俊亮(1933－),男,教授,博士生导师,中国科学院院士,中国工程院院士,主要研究领域为网络智能化.



梅翔(1979－),男,博士生,主要研究领域为Web信息检索,Semantic Web技术.